

メタ戦略と分類器を用いた交通渋滞緩和

堤田 恭太¹ 伊藤 秀剛¹ 松林 達史¹ 戸田 浩之¹

概要：本研究では、メタ戦略と分類器を組合せて用いることで、渋滞を緩和する交通信号の制御パラメータを効率的に探索する最適化手法を提案する。既知の良い指標値を得られるパラメータを正例、そうでないパラメータを負例として識別する分類器を事前に学習しておき、メタ戦略で生成したパラメータをその分類器によって判別させることで良い指標値となるパラメータを絞り込むことができ、パラメータの探索を効率化することができる。実験では、199 交差点からなる実交通環境を再現した交通シミュレーションのシナリオを用いて、1,499 の信号制御パラメータを渋滞損失時間が削減するように最適化させ、提案法は需要率に基づく算出方法である Webster の実験式によるパラメータと比べて-18.3%の渋滞緩和効果となり、代表的なメタ戦略の手法である遺伝的アルゴリズムや、分類器を用いる最新の手法と比べても効率的に最適化できたことを示す。

Traffic Signal Optimization for Reducing Congestion via Metaheuristics and Classifications

KYOTA TSUTSUMIDA¹ HIDETAKA ITO¹ TATSUSHI MATSUBAYASHI¹ HIROYUKI TODA¹

1. はじめに

国内外を問わず、都市の交通渋滞は深刻な社会的課題である。平成 24 年度プローブデータを用いた国土交通省の試算^{*1}によれば、一人あたりの年間渋滞損失時間は約 40 時間、乗車時間(約 100 時間)の約 4 割に相当すると報告されており、人々の生活が不便になるだけでなく、膨大な経済損失にも繋がっている。

こうした交通渋滞の緩和には様々なアプローチが存在するが、道路の拡幅工事のようなハードウェア的な解決策の採用は難しいことが多い。そのため、リアルタイムの混雑状況や将来の混雑の予報を道路情報板等を用いて配信して利用者に迂回を促したり、交通信号(以下、信号)の制御パラメータを渋滞が緩和するように最適化したり、といった ITS(Intelligent Transportation Systems)によるソフトウェア的なアプローチが広く行われている^{*2}。その中でも信号の

制御最適化は、研究面でも実応用面でも古くから幅広く行われており [14], [18], 実フィールドでの実証実験においても成果が報告されている [17]。

信号制御最適化の方法として、車両や信号をエージェントとして混雑状況や相互作用を模擬するマルチエージェントシミュレーションを用い、その渋滞損失時間などの指標値を改善するように最適化する方式がある。この時、指標値の関数が微分可能な関数であれば、その最適化に勾配法等を用いることができるが、実際には指標値の取得時にノイズが含まれたり、微分不可能な関数であったりするため、適用が困難もしくは不可能であることが多い。そのため、微分に関する情報を用いない最適化 (Derivative Free Optimization, DFO) の手法である遺伝的アルゴリズム (Genetic Algorithms, GA)[3] 等のメタ戦略が用いられている [2], [18], [19]。GA では、パラメータのランダムな変更(突然変異)や、良い指標値が得られるパラメータの要素を交換する操作(交叉)により、次に評価を行う新しいパラメータ(以下、探索点)を生成してその評価を行なうが、常に良いパラメータが生成されるとは限らない。そのため、シミュレーション上でパラメータを評価する回数(探索回数)が膨大となることがあり、シミュレーションの計算コ

¹ 日本電信電話株式会社 NTT サービスエボリューション研究所
〒 239-0847 神奈川県横須賀市光の丘 1-1

^{*1} 国土交通省 - 基本方針に関するデータ集 (p.4)
<http://www.mlit.go.jp/common/001067075.pdf>

^{*2} ITS Japan - ITS 全体構想 第 3 章 日本の ITS の現状 <http://www.mlit.go.jp/road/ITS/j-html/5Ministries/3.html>

ストが高い場合には探索回数を減らす工夫が必要になる。

一方、こうした DFO の問題に対して Yu ら [12], [13] は、既知の良い指標値を得られるパラメータを正例、そうでないパラメータを負例として識別する分類器を構築し、分類結果が正例となる集合から重点的にサンプリングを行うことで、効率的に最適化を行う方法を提案し、人工評価関数等のタスクでメタ戦略手法等と比べて効率的にパラメータを最適化できたことを示した。

そこで本研究では、メタ戦略 (GA) と分類器を組合せて用い、GA で生成した信号の制御パラメータを、既知の良い指標値を得られるパラメータを正例、そうでないパラメータを負例として識別する分類器を事前に学習し、その分類器によって探索の要否を判別させて絞り込むことで、効率的に渋滞を緩和するパラメータを探索する最適化手法を提案する。実験では、交通シミュレータ SUMO (Simulation of Urban MObility) [8] 上で、ルクセンブルク市の実交通環境を再現したデータセット [1] を用いた定量評価を行い、GA や、分類器のみを用いる最新のパラメータ最適化手法と比較して提案法の有効性を示した。また、提案法の分類器の学習データとして、次のような 2 つのユースケースを想定して異なる分類器を用い、需要率に基づく一般的な算出方法と比べて、提案法 a は -15.6%、提案法 b は -18.3% の渋滞緩和効果となり、いずれも比較手法と比べて効率的に最適化を行えることを示した。

- a) 平日 8 時台が常に混雑している、といった、過去に試行したパラメータの効果が既知の交通環境を改善する場合 (提案法 a)
- b) 時間経過などで交通環境が変化し、現在のパラメータが合わなくなったので見直したい、といった、パラメータの効果が未知の新しい交通環境を改善する場合 (提案法 b)

本研究の主な貢献は次の通りである：

- 分類器のみを用いる最新のパラメータ最適化手法 [12], [13] の改良として、GA で生成した信号の制御パラメータを、事前に学習した分類器によって探索の要否を判別してさらに絞り込むことにより探索を効率化する手法を提案した。
- 実交通環境を再現した広域のシミュレーション環境での評価実験を行い、交通工学的な知見に基づく手法や、GA、分類器のみを用いる最新の最適化手法 [5], [13] と比べ、提案法が優れることを示した。
- 分類器の学習データとして、ユースケースに応じて 2 種類の学習データを用いる方法を提案し、学習データに含まれない時間帯のパラメータを用いて分類器の学習を行った場合 (提案法 b) にも探索が効果化されることを示した。

また、本稿の構成は以下の通りである。2 章では、交通信号の制御方式、制御パラメータといった背景知識について述べ、最後に本研究のフォーカスと最適化するパラメータについて述べる。3 章では、信号制御の最適化と、シミュレーションを用いた最適化、遺伝的アルゴリズムなどのメタ戦略などの関連研究について述べる。4 章では、提案法について述べる。5 章では、シミュレーションの条件や各手法の設定などの評価実験について述べる。6 章では実験の結果と考察について述べる。7 章では、本研究のまとめを述べる。

2. 信号制御の方式とパラメータ

信号の基本的な役割は、交差点の同一平面上の交通流について、青、黄、赤の灯火によって当該交差点を通行する権利 (通行権) を時間的に分離することである。これにより接触事故を減らすことができ、円滑な交通を実現する。

本章では、交通信号の背景知識として、制御方式、信号現示、制御パラメータについて述べ、最後に、本研究のフォーカスと最適化する信号パラメータについて述べる。

2.1 信号制御の方式

信号の制御方式は、制御するパラメータの設定方式の観点で定周期制御と交通感応制御の大きく 2 種類に分類される [7], [15]。

定周期制御は、あらかじめ制御パラメータを作成しておき、時間帯や曜日等に応じて適用するパラメータを切り替えることで実現 (多段定周期制御) されている。例えば、交通量が多く混雑する時間帯には、信号待ちによる交通渋滞が発生しないように後述する有効青時間長の比較的長いパターンを設定し、夜間などの交通量が少ない時間帯には、信号待ちの時間が少なくなるように有効青時間長の短いパターンを設定する、といった制御が行われる。

交通感応制御は、車両感知器を用いて制御を行う方式であり、交差点に接近する車を検出して信号待ちの時間を減らしたり、交通量の増減に応じて青信号の表示時間を増減する。特に、交通需要が飽和していない (渋滞などの混雑が少ない) 場合に効果的であることが知られており、英国の SCOOT [6] などが代表的である。

2.2 信号現示と有効青時間長

信号によって交差点に対して与えられる通行権の組合せを信号現示といい、その信号現示で動作する時間の長さを有効青時間長という。図 2.2 に、一般的な十字交差点における信号現示と有効青時間長の例を示す。矢印の始点と終点は、それぞれ交通流の交差点への進入と進行方向を表し、信号現示 1 では、南北方向の直進と右折ができ、青信号が表示される有効青時間長は 40 秒である。続いて、信号現示 2 では 6 秒間の左折専用の矢印信号が灯火する。同様に、

信号現示 k	1	2	3	4
有効青時間長 d_k [秒]	40	6	30	6

図1 十字交差点での信号現示と有効青時間長 d_k の例

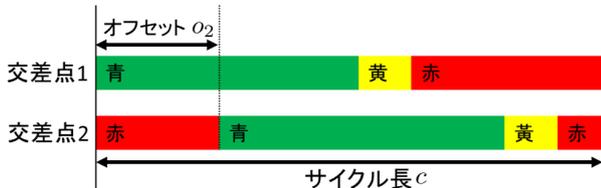


図2 サイクル長 c と交差点2のオフセット o の例

信号現示 3,4 は東西方向の進入路に対する動作を表す。

2.3 信号パラメータ

信号の制御パラメータ (以下, 信号パラメータ) には, 有効青時間長 d として現示に対応する秒数として表す他にも, サイクル長 c , スプリット g , オフセット o と呼ばれる3種類の制御パラメータを用いて表す方法がある [7], [15]. 本節ではそれらと図2での例について述べる。

サイクル長 c とは, 1つの信号 (灯器) の表示が青, 黄, 赤と1巡するのに要する時間のことであり, 当該交差点の全ての有効青時間長 d と, 現示の切替え時に表示される黄の表示時間や全灯器が赤になるクリアランス時間を含めた合計時間となる. 図2.2の例では, 黄色時間 Y が固定値4秒, クリアランス時間はないとしたとき, 現示の数は4つあるので, サイクル長 $c = \sum_k^4 (d_k + Y) = 98$ [秒] となる.

スプリット g とは, 各信号現示の有効青時間長が, 全有効青時間長の和に占める割合である. 図2.2の信号現示1の例では, スプリット g_1 は, $g_1 = d_1 / \sum_k^4 d_k \cong 0.357$, 約36%となる.

オフセット o とは, 信号現示の開始時刻をずらすためのパラメータであり, 複数の交差点を信号による停止なく通過できるように導入される. 特に, 複数の交差点の相対的な時間差で表される場合に相対オフセットといい, 基準となる時間との差として表される場合に絶対オフセットと呼ばれる. 本研究では, この絶対オフセットを扱い, 以降は単にオフセットと呼ぶ. 図2にオフセットの例を示す. 交差点1, 2間の移動にかかる時間が15秒程度の時, 絶対オフセットの秒数として交差点1のオフセット o_1 に0秒, 交差点2にオフセット o_2 秒を15秒程度に設定することで, 交差点1を青信号の間に出発した車が, 交差点2が青信号の間に通過でき, 信号による停止回数や信号待ちの時間を減らすことができる.

2.4 本研究のフォーカスと最適化する信号パラメータ

最後に, 本研究のフォーカスと最適化する信号パラメータについて述べる.

本研究では, “日常的に混雑している午前8時台のパラメータを見直す”, といった一般道路での典型的な交通渋滞の緩和やそのパラメータの策定を支援することを主なターゲットとするため, 2.1節で示した多段定周期制御のパラメータ最適化に取り組む.

最適化するパラメータの種類は, いずれも正の整数値であるオフセットと有効青時間長とし, 交差点 i のオフセット $o_i \in \mathbb{Z}_+$, 交差点 i の現示数が K_i のとき, K_i 次元の有効青時間長のベクトル $\mathbf{d}_i \in \mathbb{Z}_+^{K_i}$ からなるベクトルを, 交差点 i の信号パラメータ $\mathbf{s}_i = (o_i \mathbf{d}_i)$ とする.

また, 本研究は複数の交差点を最適化対象として扱う. 対象すべての交差点の集合 \mathcal{I} について結合した, $D_{\mathcal{I}}$ 次元の正の整数からなる全信号パラメータのベクトルを, 単に信号パラメータ $\mathbf{s} = (\mathbf{s}_1 \dots \mathbf{s}_{|\mathcal{I}|}) \in \mathbb{Z}_+^{D_{\mathcal{I}}}$ と表す. 以降, 最適化する信号パラメータとしてこれを用いる.

3. 関連研究

本章では, 需要率に基づくパラメータ算出の方法である Webster の実験式 [14] と, シミュレーションを用いた最適化の枠組みについて述べ, 続いて, シミュレーションを用いた最適化でよく用いられている遺伝的アルゴリズム [3], 分類器を用いた最新の最適化手法 [12], [13] について述べる.

3.1 需要率に基づくパラメータ算出 (Webster の実験式)

英国道路研究所の Webster は, 交差点に進入する交通流がポアソン過程に従うことを仮定したシミュレーション実験を行い, 各交差点での遅れ時間を最小にするパラメータを算出する近似式を示した [14]. これは Webster の実験式と呼ばれ, 単独の信号交差点に対する定周期制御のパラメータ算出によく用いられる方法である. 下記の Webster の実験式により, 交差点 i のサイクル長 c_i , 交差点 i の k 番目の信号現示のスプリット $g_{i,k}$ と有効青時間長 $d_{i,k}$ を求めることができる.

$$c_i^{(WS)} = \frac{1.5L_i + 5}{1 - \lambda_i}, \quad (1)$$

$$g_{i,k}^{(WS)} = \frac{\lambda_{i,k}}{\lambda_i}, \quad (2)$$

$$d_{i,k}^{(WS)} = (c_i^{(WS)} - L_i)g_{i,k}^{(WS)}. \quad (3)$$

ここで, L_i は i 番目の交差点における損失時間の長さを表し, 青信号が表示されてからドライバーが発進するまでの発進損失や交差点に残った車が捌けるまでのクリアランス損失など, 信号現示の切替えに付随して増加するパラメータである. 本研究では, 交差点 i の信号現示の数 K_i を用い, 現示の切替に付随するコスト R を3として, $L_i = K_i * R$ と算出して用いた. また, λ_i は交差点 i の需要率, $\lambda_{i,k}$ は

その k 番目の信号現示の需要率である。需要率は、当該道路の交通量の計画値 (一般車に換算して 1250[台数/車線/時] など) に対する交通量の割合を表し、実用的には、交差点 i の需要率 λ_i が 0.9 を超えると極端に大きなサイクル長が算出されることが知られている [16]。そのため、本研究では計画値を 1250[pcu/車線/時]、交差点の需要率の上限を 0.85 とし、 $\lambda_i = \min(0.85, \lambda_i)$ とし算出したものを用いた。

この様に、Webster の実験式を用いることで、道路や現示に対する混雑の度合いである需要率を考慮して適したパラメータを算出することができる。しかしながら、

- 本式では、オフセットの算出はできないこと
- 近接する信号交差点の影響等で、必ずしも車は実験式で仮定するポアソン過程に従わない到着となること [7], [11]
- 車の平均速度が混雑時には低下する現象 [4] など、他の車の影響が考慮されていないこと

といった理由から、実用的には車同士の相互作用などの複雑な現象を再現した交通シミュレーションによる評価やパラメータの最適化も用いられている。

3.2 シミュレーションを用いた信号パラメータの最適化

前述の交通環境における車同士の相互作用のように、複雑な現象の影響を考慮してパラメータを最適化する場合、それらをエージェントとするマルチエージェントシミュレーションを用いて計算機上で現象を再現、観察できるようにし、エージェントの動作パラメータを最適化するアプローチがある [10]。信号パラメータの最適化に交通シミュレータを用いる場合は、車や信号をシミュレーション上のエージェントとし、信号 (エージェントの動作) パラメータをシミュレーション上で混雑度合いを表す指標値が改善するように最適化を行う [2], [18], [19]

実行可能な信号パラメータの集合を \mathcal{S} 、信号パラメータを $s \in \mathcal{S}$ 、指標値を $l \in \mathbb{R}$ 、信号パラメータを変数として指標値を算出するシミュレーションを目的関数 $f: \mathcal{S} \rightarrow \mathbb{R}$ とし、その最適解である信号パラメータ s^* を探索する最小化問題 $s^* = \arg \min_{s \in \mathcal{S}} f(s)$ が、本研究で扱うシミュレーションを用いた信号パラメータを最適化する問題となる。

3.3 シミュレーションとメタ戦略 (GA) を用いた最適化

シミュレーションを用いた最適化では、目的関数であるシミュレーションでの評価値にランダムなノイズが含まれたり、評価値の関数が不連続的であったりといった性質があるため、目的関数の微分に関する情報を用いる勾配法などが適さないことが多い。そのため、関数の微分に関する情報を用いない最適化 (Derivative Free Optimization, DFO) の手法である遺伝的アルゴリズム (Genetic Algorithms, GA)[3] 等のメタ戦略がよく用いられている [2], [18], [19]。

Algorithm 1 メタ戦略 (GA) による最適化

Input:

- f: 最小化する目的関数 (シミュレーションによる評価);
- Sampling: \mathcal{S} からのサンプリング処理;
- Selection: 目的関数で評価し、淘汰する処理;
- $\mathcal{G}_t \in \mathcal{S}$: t 世代目の個体集合;
- $\pi^{(cx)} \in [0, 1]$: 個体の要素を一様交叉する確率;
- $\pi^{(mut)} \in [0, 1]$: 個体の要素を突然変異する確率;
- $T \in \mathbb{N}_+$: イテレーション数;
- $G \in \mathbb{N}_+$: 世代の個体数;
- $X \in \mathbb{N}_+$: 交叉させる個体の組数;
- $M \in \mathbb{N}_+$: 突然変異させる個体数。

Procedure:

- 1: $\mathcal{G}_1 = \text{Sampling}(\mathcal{S})$; // ランダムにサンプリングして初期化
- 2: **for** $t = 1, \dots, T$ **do**
- 3: $\mathcal{G}_t = \text{Selection}(\mathcal{G}_t)$; // 評価して、淘汰
- 4: $\mathcal{G}_{t+1} = \emptyset$ に初期化;
- 5: **for** $i = 1, \dots, X$ **do**
- 6: \mathcal{G}_t からランダムに 2 個体 $\{s_{t,i,a}, s_{t,i,b}\}$ を選択;
- 7: 選択した 2 個体の要素を、 $\pi^{(cx)}$ で交換して、新たな探索点 $\{s'_{t,i,a}, s'_{t,i,b}\}$ を生成; // 交叉
- 8: $\mathcal{G}_{t+1} = \mathcal{G}_{t+1} \cup \{s'_{t,i,a}\}$; // (次の世代に追加)
- 9: $\mathcal{G}_{t+1} = \mathcal{G}_{t+1} \cup \{s'_{t,i,b}\}$; // (次の世代に追加)
- 10: \mathcal{G}_t から選択した 2 個体を除く;
- 11: **end for**
- 12: **for** $j = 1, \dots, M$ **do**
- 13: \mathcal{G}_t から j 番目の個体 $s_{t,j}$ を選択;
- 14: 個体の要素を $\pi^{(mut)}$ でランダムに更新し、新たな探索点 $s'_{t,j}$ を生成; // 突然変異
- 15: $\mathcal{G}_{t+1} = \mathcal{G}_{t+1} \cup \{s'_{t,j}\}$; // (次の世代に追加)
- 16: \mathcal{G}_t から選択した個体を除く;
- 17: **end for**
- 18: **end for**
- 19: $\tilde{s} = \arg \min_{s \in \mathcal{G}_T} f(s)$;
- 20: **return** \tilde{s}

Alg. 1 は、シミュレーションとメタ戦略 (GA) を用いる最適化の処理フローの例である。GA で最適化するパラメータを個体と呼び、個体の要素をランダムな変更 (突然変異) や、2 個体の組について一部要素を交換する操作 (交叉) を行って新たに評価する個体 (探索点) として得、それらを目的関数で評価し、評価値の良い個体を残す (淘汰)、という手続きを指定回数繰り返すことでパラメータを最適化する。GA によって信号パラメータの最適化を行う場合、信号パラメータ s を個体として用いる。

3.4 分類器を用いる最適化 (CBO)

Yu ら [12] は、既知の良い評価値を得られるパラメータの集合と、そうでないパラメータとを識別する分類器を構築し、目的関数での評価を行わずに分類器の判定結果が正となる集合と、パラメータの定義域や制約を満たす実行可能解の集合から重み付きでサンプリングすることによってパラメータを生成することで、効率的に最適化を行う方法を提案した。以下、この手法を Yu ら [13] にならって (Classification-based optimization, CBO) と呼ぶ。また、分

Algorithm 2 分類器を用いる最適化 (CBO)[12], [13]**Input:**

- f: 最小化する目的関数 (シミュレーションによる評価);
- bc: 分類器 h の構築処理 (分類器の学習);
- Sampling: \mathcal{S} からのサンプリング処理;
- $\mathcal{G}_t \in \mathcal{S}$: t 世代目の探索点集合;
- $\mathcal{S}^{(h=+1)}$: 分類器が正と判定したパラメータの集合;
- α : 分類の正負の閾値;
- $\pi \in [0, 1]$: ランダムサンプリングの確率;
- $T \in \mathbb{N}_+$: イテレーション数;
- $G \in \mathbb{N}_+$: 探索点の数 (=GA の世代の個体数).

Procedure:

- 1: $\mathcal{G}_1 = \text{Sampling}(\mathcal{S})$; // ランダムにサンプリングして初期化
 - 2: $\tilde{s} = \arg \min_{s \in \mathcal{G}_0} f(s)$ に初期化;
 - 3: **for** $t = 1, \dots, T$ **do**
 - 4: $y_k = \text{sign}[\alpha - f(s_k)]$ とし, 分類器の学習データ $\mathcal{T}_t = \{(s_1, y_1), \dots, (s_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$ を構築;
 - 5: $\mathcal{G}_t = \emptyset$ に初期化;
 - 6: **for** $g = 1, \dots, G$ **do**
 - 7: $h_t = \text{bc}(\mathcal{T}_t)$;
 - 8: $s_g = \begin{cases} \text{Sampling}(\mathcal{S}^{(h_t=+1)}) & (\text{w.p. } 1 - \pi) \\ \text{Sampling}(\mathcal{S}) & (\text{w.p. } \pi) \end{cases}$
 - 9: $\mathcal{G}_t = \mathcal{G}_t \cup \{s_g\}$;
 - 10: **end for**
 - 11: $\tilde{s} = \arg \min_{s \in \mathcal{G}_t \cup \{\tilde{s}\}} f(s)$;
 - 12: **end for**
 - 13: **return** \tilde{s}
-

分類器を構築する際, パラメータの解空間からのサンプリングを既知の良いパラメータを含む座標軸上でを行い, その範囲も徐々に狭めていく RACOS[13] を提案した. 実験では, Sphere 関数や Ackley 関数等の人工評価関数や, クラスタリングのパラメータを最適化するタスクにおいて, メタ戦略手法等と比べて効率的にパラメータを最適化できたことが示されている.

Alg. 2 に, 分類器を用いる最適化 (CBO) の手順を示す. 分類器 h_t の学習データ \mathcal{T}_t を閾値 α に応じて構築し (Alg. 2 の 4, 7 行目), 分類器 $h_t: \mathcal{S} \rightarrow \{+1, -1\}$ の分類結果が $+1$ のパラメータの集合 $\mathcal{S}^{(h=+1)}$ と, 探索範囲からのランダムサンプリングによって探索点を生成, 評価を行って, 評価値の良いパラメータがあれば \tilde{s} を更新し, 上限 T 回までこの手続を行って最適化を進める.

4. 提案法

本章では, メタ戦略 (GA) と分類器を用いる最適化である提案法について述べる. Alg. 3 に提案法の処理手順を示す.

3.3 節で述べた GA の計算では, シミュレーションによって評価を行う新たなパラメータ (探索点) を, 交叉や突然変異と呼ばれる操作によって生成する. しかしながら, 常に良いパラメータが生成されるとは限らないため, 特にシミュレーションを用いた最適化のような 1 回の評価に多くの計算資源を必要とする場合には評価回数を減らす工夫が必要である. そこで提案法では, GA の交叉や突然変異

の操作で生成したパラメータを, 事前に学習した分類器によって探索の要否を判別させて, 更に絞り込むことで探索を効率化する (Alg. 3 の 13–15 行目).

また, Yu ら [12], [13] による CBO では, パラメータの生成が, 初期化時と確率 π での探索範囲からのランダムサンプリングに依存している (Alg. 2 の 1 行目と 8 行目). ランダムサンプリングでは, 分類器 h の学習時に正例となる良い解も得られにくいいため, Yu らも RACOS[13] においては探索範囲を動的に縮小する工夫を行っている. そこで提案法では, RACOS のようにサンプリングする範囲を縮小するのではなく, サンプリング機能としてランダムサンプリング (ランダムサーチ) より効率的に最適化が可能な GA を用いて効率化を図る. (Alg. 3 の 12 行目) 加えて CBO では, 探索ごとに分類器 h の学習があり (Alg. 2 の 4 行目と 7 行目), 最適化パラメータが高次元な場合には, 分類器 h の十分な分類精度に要するデータ量も大きくなるため, この学習時間が無視できないものとなる. そこで提案法は, 次節で述べるようにユースケースに応じて事前に分類器の学習を行っておく (Alg. 3 の 1–7 行目).

4.1 ユースケースと分類器の学習データ

1 章で述べたように, 2 種類のユースケースを考え, それに応じた学習データを用いて提案法 a, 提案法 b とする. 提案法 a は, 平日 8 時台が常に混雑しているといった, 過去に試行したパラメータの効果が既知の交通環境の改善を想定し, パラメータの最適化を行う交通環境が 8 時台のとき, 学習データの時間帯 \mathcal{J} も $\mathcal{J} = \{8\}$ とする.

提案法 b は, 時間経過などで交通環境が変化し, 現在のパラメータが合わなくなったので見直したい, といった, 新しい交通環境の改善を想定し, パラメータの最適化を行う交通環境の 8 時台以外のデータ $\mathcal{J} = \{0, \dots, 23\} \setminus \{8\} = \{0, \dots, 7, 9, \dots, 23\}$ を用いる.

4.2 複数時間帯を扱う学習データと分類器の学習

提案法 b のように, 学習データに複数の時間帯のデータを用いる場合, 閾値 α は, 指標値を基準値としても用いるのではなく, 当該時間の交通環境において良いパラメータであるか否か, を表現した閾値である必要がある. そこで, 基準となる解として Webster の実験式によるパラメータを用い, その相対的な改善度合いと閾値を比較し, $y_{j,b} = \text{sign}[\alpha - (f(s_{j,b})/f(s_j^{(WS)}))]$ とし, 正例負例のラベル y を付加する (Alg. 3 の 4 行目). また, $\{(s_{j,b}, \theta_j), y_{j,b}\}$ は, 特徴ベクトル $(s_{j,b}, \theta_j)$ とその正負のラベル y からなる学習データである. ここで, θ_j は, 時間帯 j における交通環境を表すベクトルであり, これによって様々な交通環境に応じて信号パラメータ s の良し悪しを分類器が判別できるようになることを期待している. 本研究では, Webster の実験式によって算出される有効青時間長 d と交差点のサ

Algorithm 3 [提案法] メタ戦略と分類器を用いる最適化

Input:

GASampling: Alg. 1 の 2–18 行目) によるサンプリング処理;
 \mathcal{J} : 学習データの時間帯の集合;
 \mathcal{B}_j : 時間帯 j における蓄積された学習データの集合;
 θ_j : 時間帯 j における交通環境のベクトル;
 $\Phi^{(GA)}$: Alg. 1 で用いる GA のパラメーター式;
 $\Phi^{(CBO)}$: Alg. 2 で用いる CBO のパラメーター式;

Procedure:

```

1:  $\mathcal{T} = \emptyset$  に学習データを初期化;
2: for  $j \in \mathcal{J}$  do
3:   for  $b, \dots, |\mathcal{B}_j|$  do
4:      $y_{j,b} = \text{sign}[\alpha - (f(s_{j,b})/f(s_j^{(WS)}))]$  とし,
        $\mathcal{T} = \mathcal{T} \cup \{(s_{j,b}, \theta_j), y_{j,b}\}$  として,
       分類器の学習データを構築;
5:   end for
6: end for
7:  $h = \text{bc}(\mathcal{T})$ ; // ただ 1 つの分類器  $h$  を構築
8:  $\mathcal{G}_1 = \text{Sampling}(\mathcal{S})$ ; // ランダムにサンプリングして初期化
9: for  $t = 1, \dots, T$  do
10:   $\mathcal{G}_{t+1} = \emptyset$  に初期化;
11:  while  $|\mathcal{G}_{t+1}| < G$  do
12:     $s = \text{GASampling}(\mathcal{G}_t, \Phi^{(GA)})$ ;
        // GA でサンプリングしたパラメータを分類器に従って追加
13:    if  $h(s) > 0$  then
14:       $\mathcal{G}_{t+1} = \mathcal{G}_{t+1} \cup \{s\}$ ;
15:    else
16:       $E = E + 1$ ; // 集計用にパラメータの除外数をカウント
17:    end if
18:  end while
19:   $\tilde{s} = \arg \min_{s \in \mathcal{G}_t \cup \{\tilde{s}\}} f(s)$ ;
20: end for
21:  $\hat{s} = \arg \min_{s \in \mathcal{G}_T} f(s)$ 
22: return  $\hat{s}$ 

```

イクル長 c が、混雑度を表す需要率を反映したものであることを利用して、 $\theta = (d^{(WS)} c^{(WS)})$ として用いた。

5. 実験

本章では、シミュレータと検証用のデータセット等の実験条件、比較手法について述べる。

5.1 交通シミュレータ SUMO[8]

本研究では、交通シミュレータとして SUMO(Simulation of Urban MObility)[8] を用いた。車両の走行速度等のパラメータは既定値とし、車両の経路選択規則等はルクセンブルクシナリオ [1] に従うこととした。

5.2 検証データ：ルクセンブルクシナリオ (LuST)

ルクセンブルクシナリオ (Luxembourg SUMO Traffic Scenario, LuST)[1] は、ルクセンブルク大の Codecá らによって整備された交通シミュレータ SUMO 用のシミュレーションシナリオであり、信号制御の検証を含む様々な用途で用いることができる。特に次の 3 点の再現がなされており、本研究の評価実験に適したデータとなっている。

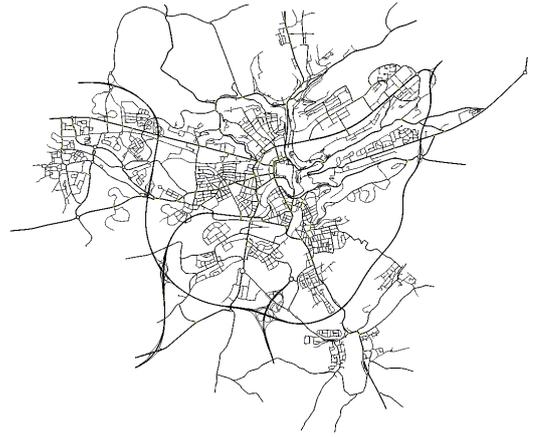


図 3 ルクセンブルクシナリオ [1] の道路ネットワーク

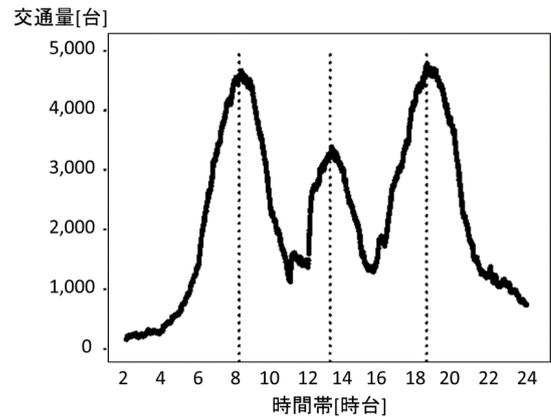


図 4 ルクセンブルクシナリオ [1] の時間帯別の交通量

表 1 LuST データに含まれる項目

項目	
範囲 [km ²]	155.95
道路の全長 [km]	931.11
交差点数	4,473
信号交差点数 [I]	203 (内, 199 を利用)
信号パラメータの次元数 s	1,499
内, 有効青時間長 d の数	1,300
内, オフセット o の数	199

- 多数の信号交差点を含む広域な実交通環境 (図 3)
- 時間帯毎の交通量の変化 (図 4)

5.2.1 LuST の統計情報と信号パラメータベクトル s

表 1 に、LuST データに含まれる道路ネットワークや交差点数といった本研究に関わる項目の統計情報を載せた。その内、最適化する信号パラメータのベクトル $s \in \mathbb{Z}_+^{|I|}$ は、信号交差点数 $|I|$ が、常に青表示となる信号現示 1 つのみの交差点等を除いて 199 箇所を用い、交差点毎に付与されるオフセット o の 199 件、有効青時間長 d の 1,300 件を合わせて、最終的に $|s| = 1499$ 次元の信号パラメータベクトル $s \in \mathbb{Z}_+^{1499}$ を用いた。

5.2.2 シミュレーション対象の時間帯

本研究では、LuST データのうち、朝の通勤時間帯で混

雑が見られる(図4)午前8時からの30分間を実験用の交通環境として用いた。

5.3 評価指標値(渋滞損失時間)

信号パラメータを評価する指標値としては、渋滞損失時間や信号交差点での停止回数などがある[15]が、本研究では渋滞損失時間を用いた。渋滞損失時間とは、車がある区間を通過するのに要した時間から、制限速度などの基準となる速度で同区間を通過した場合の所要時間を引いた値の全走行車両に対する総和であり、経済的な改善効果への換算も容易であることからITS分野では施策の効果を把握する目的で広く用いられている。

5.4 比較手法

本節では、比較手法と提案法a、提案法bについて述べる。大域最適解が得られる保証のないパラメータの最適化は、得られる解が初期解の影響を受けることがある。そのため、いずれの手法も共通して3.1節のWebsterの実験式で得られた有効青時間長のベクトル $\mathbf{d}^{(WS)}$ とオフセットのベクトル $\mathbf{o}^{(WS)}$ からなる信号パラメータ $\mathbf{s}^{(WS)} = (\mathbf{d}^{(WS)} \mathbf{o}^{(WS)})$ を、最適化の初期解として用いた。また、探索範囲も各手法で共通とし、交差点 i のオフセット o_i は、Websterの実験式(詳細は、3.1節)で求めたサイクル長 $c_i^{(WS)}$ のオフセット率で $\pm 50\%$ までとした。また、交差点 i の k 番目の現示に対する有効青時間長 $d_{i,k}$ も同様にWebsterの実験式によって求めた $d_{i,k}^{(WS)}$ の $+50\%$ を探索範囲の上限として用い、1秒から $\lceil d_{i,k}^{(WS)} \times 1.5 \rceil$ 秒を探索の範囲とした。

- ランダムサーチ (RS_Webster)
パラメータを探索範囲でランダムに変更する方法。
- 遺伝的アルゴリズム (GA)[3]
3.3節で述べたGAによる方法。各世代の個体数 G を100個体、 t 世代目の個体の集合 \mathcal{G}_t に対して、7割の個体($X = 35$ 組, 70個体)を交叉、残りの3割の個体($M = 30$ 個体)に突然変異の操作を行なった。淘汰の処理(Selection)は、5段のトーナメント方式を用いた。
- 分類器を用いる手法 (RACOS)[13]
3.4節で述べた分類器を用いる最新の最適化手法。ハイパーパラメータとなる閾値 α は事前にデータ量の10%, 30%, 50%の場合について探索効率を調べ、最もよい値となった10%となる値を用い、 π も同様に0.1, 0.3, 0.5の場合を調べ0.1を用いた。
- 分類器を用いる手法 (SRACOS)[5]
RACOSで用いる学習データの内、探索中に得られる良い評価値の解と、悪い評価値の解とを探索が進む毎にそれぞれ入れ替えることで発展させた最新の最適化手法。 α , π はRACOSと同様とした。
- 提案法a(分類器の学習データ = 交通環境)

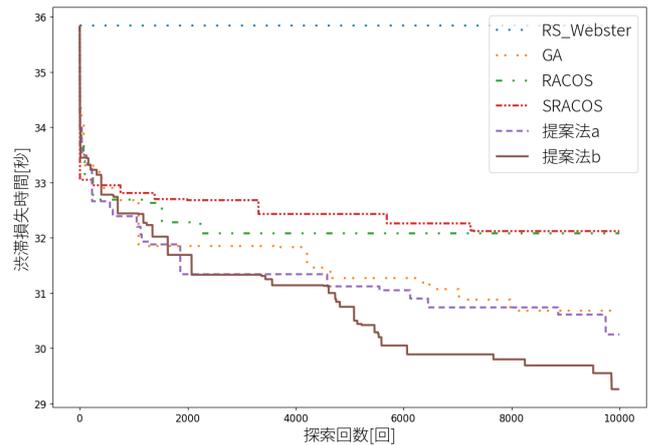


図5 各手法の最適化結果

4章で述べた提案法であり、分類器の学習データに、最適化する交通環境と同一の8時台のデータを用いた手法。 $\Phi^{(GA)}$ はGAと同一。閾値 α は上位10%が正例となる0.99, 学習データ $|T|$ は約6,000件を用いた。

- 提案法b(分類器の学習データ \neq 交通環境)
提案法aと異なり、分類器の学習データに交通環境の8時台以外のデータ $\mathcal{J} = \{0, \dots, 7, 9, \dots, 23\}$ を用いた手法。閾値 α は上位10%が正例となる0.97, 学習データ $|T|$ は約5,000件を用いた。

6. 結果と考察

6.1 実験結果

図5に、各手法での信号パラメータの最適化結果を示す。横軸は、シミュレーションによって評価を行った回数(探索回数)であり、縦軸は、低い値が良い評価値である渋滞損失時間(5.3節)である。図から、提案法a、提案法bは比較手法と比べて効率的に良いパラメータを探索したことが分かる。また、GAとの比較により、提案法の分類器による絞り込みが有効であることが分かるが、提案法aとGAとの差が小さい。これは上位10%を正例とした分類器の学習に用いる閾値が大きかったためと考える。

表6.1は、各手法の渋滞損失時間の探索回数 10^4 回までの最小値と、提案法の分類器による除外数 E である。最も良い効果となった評価値を太字で示しており、提案法bが29.26で最も良い渋滞緩和効果となった。これはRS_Websterを基準として-18.3%の削減効果である。また除外数 E は、分類器によって探索不要とされたパラメータのカウント数(Alg.3の16行目)である。提案法bでは10,000件を探索する間に、4,253件のパラメータを探索不要として除外できており、提案法aと比べて効率化に寄与したことが分かる。

6.2 GAによる効率化の分析

図6に、ランダムサーチによる t 回目までの最小値

表 2 渋滞損失時間の最小値と、分類器によって除外された解の数

methods	10 ⁴ 回の最小値	分類器による除外数 E
RS_Webster	35.84	0
GA	30.68	0
RACOS	32.08	(サンプリング時に考慮)
SRACOS	32.12	(サンプリング時に考慮)
提案法 a	30.25	2,259
提案法 b	29.26	4,253

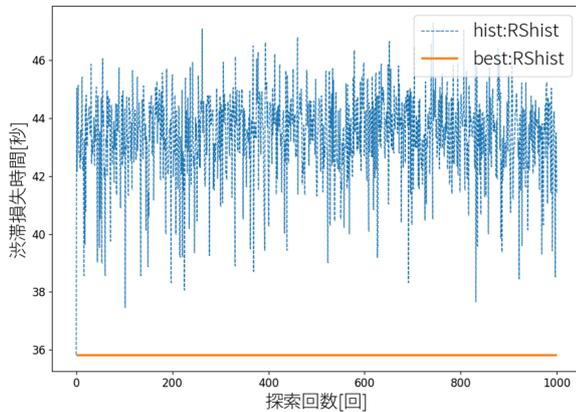


図 6 ランダムサーチ (RS_Webster) による、 t 回目までの最小値 (best:RS_Webster) と、 t 回目の評価値 (hist:RS_Webster)

(best:RS_Webster) と、 t 回目の評価値 (hist:RS_Webster) を示した。この結果から、Webster の実験式による初期解から探索範囲内でランダムにパラメータを更新すると評価値が悪化しやすいことが伺える。またこの事は、ランダムサーチや、探索範囲からのサンプリングを行う (S)RACOS と比べて、GA によるサンプリングを行う提案法が効率的にパラメータを探索する理由の一つであると考えられる。

最後に、結果から分かることを次の通りまとめる。

- GA や最新の最適化手法を含む比較手法と比べ、提案法は探索効率で優れ (図 5)、GA との比較により、提案法の分類器による絞り込みは有効である。
- ランダムサーチの結果 (図 6) から、ランダムなパラメータの更新は探索効率を下げる可能性が高く、GA や提案法で行ったサンプリング部分の GA へ置き換えは有効である。
- 提案法 b が最も効率的に最適化を行うことができ、分類器は目的の交通環境以外のデータを用いて学習しても効果があり (Webster の実験式を基準として -18.3% の削減効果)、提案法 a のように事前に交通環境に対するパラメータの効果が把握できないような、未知の交通環境に対しても探索の効率化が可能である。

7. まとめ

本研究では、メタ戦略と事前に学習した分類器を組み合わせることで、渋滞を緩和する信号パラメータの探索を効率的に行う手法を提案した。実際の広域な都市交通環境を再現

した交通シミュレーションを用いた比較実験において、提案法は需要率に基づく一般的な算出方法と比べて -18.3% の渋滞緩和効果となり、遺伝的アルゴリズムや、分類器を用いる最新の手法と比べても効率的に最適化できた。

参考文献

- [1] Codecá, L., Frank, R., Faye, S. and Engel, T.: “Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation”, *IEEE Int. Tran. Sys. Mag.*, pp. 52–63, 2017.
- [2] García-Nieto, J., Toutouh, J. and Alba, E.: “Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization”, *IEEE Trans. on Evol. Comp.*, Vol.17, No.6, pp.823–839, 2013.
- [3] Golberg, D. E.: “Genetic Algorithms in Search, Optimization, and Machine Learning”, MA: Addison-Wesley, 1989.
- [4] Greenberg, H.: “An analysis of traffic flow.” *Operations research*, vol. 7, pp.79–85, 1959.
- [5] Hu, Y.-Q., Qian, H. and Yu, Y.: “Sequential classification-based optimization for direct policy search”. in *Proc. AAAI’17*, pp.2029-2035, 2017.
- [6] Hunt, P. B., Robertson, D. I., Bretherton, R. D. and Winton, R. I.: “SCOOT – a traffic responsive method of coordinating signals”, TRRL Laboratory Report1014, 1981.
- [7] Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., Bonneson, J., Tarnoff, P. and Urbanik, T.: “Traffic Signal Timing Manual”, U.S. Dep. of Transportation, 2008.
- [8] Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L.: “Recent development and applications of SUMO-Simulation of Urban MOBility”, *Int. Jou. On Adv. in Sys. and Mea.*, pp. 128–138, 2012. Panovski, D. and Zaharia, T.: “Simulation-based vehicular traffic lights optimization”, in *Proc. SITIS’16*, 2016.
- [9] Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. and Wang, Y.: “Review of road traffic control strategies”, in *Proc. IEEE*, Vol. 91, No. 12, pp. 2043–2067, 2003.
- [10] Stone, P. and Veloso, M.: “Multiagent systems: A survey from a machine learning perspective”, *Autonomous Robots*, Vol.8 Num.3, pp.345–383, 2000.
- [11] Xie, X.-F., Barlow, G. J., Smith, S. F., and Rubinstein, Z. B.: “Platoon-Based Self-Scheduling for Real-Time Traffic Signal Control”, in *Proc. ITSC’11*, 2011.
- [12] Yu, Y. and Qian, H.: “The sampling-and-learning framework: A statistical view of evolutionary algorithm”. in *Proc. CEC’14*, pp.149-158, 2014.
- [13] Yu, Y., Qian, H. and Hu, Y.: “Derivative-Free Optimization via Classification”, in *Proc. AAAI’16*, pp. 2286–2292, 2016.
- [14] Webster, F. V.: “Traffic Signal Settings”, Road Research Technical Paper, Great Britain Road Research Lab., 1958.
- [15] (一社) 交通工学研究会. “改訂 交通信号の手引き”, 2006.
- [16] (一社) 交通工学研究会. “平面交差の計画と設計 基礎編, 応用編”, 丸善出版, 2007.
- [17] (株)NTT データ: “中国・貴陽市でビッグデータを活用した「渋滞予測・信号制御ソリューション」のエリア拡大実験においても渋滞緩和効果を確認”, 入手先 (<http://www.nttdata.com/jp/ja/news/release/2016/112800.html>) (参照 2018/04/12).
- [18] 織田 利彦, 音喜多 亨, 津久家 智光, 橋場 加奈: “メタ戦略による TRANSYT 交通流モデルにおける交通信号制御最適化”, システム制御情報学会論文誌, Vol. 10, No. 10, pp.547-555, 1997.
- [19] 久井守, 金子 数馬: “系統交通信号の制御パラメータ連続自動調整によるリアルタイム制御の一方式”, 土木計画学研究・論文集, pp. 877–885, 2007.